

RESEARCH

Open Access

About complexity of complex networks



Alexander Goryashko^{1*} , Leonid Samokhine² and Pavel Bocharov³

“The interplay between simplicity and complexity is the heart of our subject.” — Murray Gell-Mann, *Let's call it plectics*, Complexity, Vol. 1, no. 5, 1995/96

*Correspondence:

petrovich4you@gmail.com

¹Moscow Technological Institute,
Moscow, Russia

Full list of author information is
available at the end of the article

Abstract

We derive complexity estimates for two classes of deterministic networks: the Boolean networks $S(B_{n,m})$, which compute the Boolean vector-functions $B_{n,m}$, and the classes of graphs $G(V_{P_{m,l}}, E)$, with overlapping communities and high density. The latter objects are well suited for the synthesis of resilience networks. For the Boolean vector-functions, we propose a synthesis of networks on a NOT, AND, and OR logical basis and unreliable channels such that the computation of any Boolean vector-function is carried out with polynomial information cost.

All vertexes of the graphs $G(V_{P_{m,l}}, E)$ are labeled by the trinomial $(m^2 \pm l, m)$ -partitions from the set of partitions $P_{m,l}$. It turns out that such labeling makes it possible to create networks of optimal algorithmic complexity with highly predictable parameters. Numerical simulations of simple graphs for trinomial $(m^2 \pm l, m)$ -partition families ($m = 3, 4, \dots, 9$) allow for the exact estimation of all commonly known topological parameters for the graphs. In addition, a new topological parameter—overlapping index—is proposed. The estimation of this index offers an explanation for the maximal density value for the clique graphs $G(V_{P_{m,l}}, E)$.

Keywords: Partition, Simple graph, Labeling, Algorithmic complexity, Information cost, Trinomial coefficients, Predictability, Overlapping community, Topological parameters, Connectivity, Emergency, Robustness

Introduction

The fundamental problem of the theory of complex networks is to choose the “right” measure of complexity. For the theory of logical networks (and the theory of algorithms in general), the “right” measures are based on estimates of the resources required for solving the most difficult tasks from given classes. The theory of complex networks aims to study any type of object (brain, social network, or otherwise) that is considered “complex” by definition. The more data that are collected, the better the hope of understanding the nature of a complex system—i.e., dataism, or the principle of “brute force of data”. There is an established opinion (Krioukov 2014) that “as far as the brain and other complex systems and networks are concerned, we are at a rather Ptolemaic stage, collecting the data, awaiting for Copernicus”. Is there clear representation of which data we must collect or when the process of “mining of big data” could well suffice? Could it be something like “Waiting for Godot”?¹

¹Samuel Beckett, “Waiting for Godot”.

Herein, we study the approach “from simplicity to complexity” (see (Donetti et al. 2006)) as applied to evolving complex networks, with a special emphasis on the estimation of different complexity measures: algorithmic complexity, running complexity, informational complexity and topological complexity. Although it has been empirically established that different real-world networks (brain, technological, social) have “almost the same” nontrivial topological features, it remains a mystery as to whether one type or another of graph topology is better suited for solving problems. There is a good reason to believe that methods and results of the computational complexity theory can be useful both for justifying the choice of the topology of complex networks to be designed and for understanding the real complex networks. In (Helbing et al. 2014), different cases of the “minimally invasive” principle of “chaos control” in complex networks with simultaneously extremely variable and highly predictable behavior are discussed. Designing networks that demonstrate this principle with precision is the primary goal of our paper.

For any real-world, evolving complex network, the basic stages of evolution (at least, from the algorithmic point of view) are almost identical. In the initial moment, “a set of the basic elements” comes into being that will define the future of the evolving network. As a rule, any evolving system—from genetic networks to computer networks to transportation networks—tends to utilize, as much as possible, any useful decision made at a previous stage of selection. In fact, the main principle of evolving complex networks design is scaling—at least, if the new design is working sufficiently well.

The term “complex networks” includes two different (vast and entangled) conceptions: network and complexity. Even if a network is described by a simple graph (indirect, unweighted edges containing no graph loops or multiple edges) with nodes without attributes, the problem of graph classification is challenging. The resolution of such a problem is likely impossible without more detailed research in the field of the methods of deterministic network synthesis. Recall that studying such practical problems as the creation of resilience networks calls for deeper research of regular or “almost regular” graphs, such as the families of Cayley, Cages and Ramanujan graphs. In (Donetti et al. 2006), the authors report finding topologies of deterministic “almost regular” graphs, called interwoven (entangled) graph structures. Their method (Donetti et al. 2006) requires substantial computer resources to find “almost regular” graphs, even for small numbers of graph vertices.

It is well known that graph theory serves as a theoretical basis for network design. In particular, in the seminal work (Erdős and Rényi 1960), it was established for the first time that the topological behavior of probabilistic graphs with V vertices and E edges, namely, the appearance of subgraphs of a given type (trees, cycles of given order, complete subgraphs, etc.) depends on the ratio V/E . Almost all results of (Erdős and Rényi 1960) were asymptotic, with $E \sim o(V^{1/2})$ or $E \sim \text{const}(V^{1/2})$, e.g., when the graphs have a small density. But the density of some real-world objects, for example, computer networks and transcription networks (Sorrells and Johnson 2015), is close to 1. Thus, the topological properties of deterministic graphs with overlapping cliques are of chief interest.

One of main goals of our work is to describe a new approach to the design of special classes of the deterministic networks that are both simple from the viewpoint of algorithmic complexity and entangled from the viewpoint of topological complexity. These features can be ensured by a special method of labeling network nodes in some metric space. Thus far, this approach has not commonly been used in network synthesis theory,

and at first glance, it might seem too sophisticated. In reality, this approach is no more exotic than identifying logical elements in computer networks. It should be especially emphasized that, with this approach, one uses a deterministic method for designing networks with *exactly predictable* topological parameters (at least, for the parameters being checked in numerical modeling).

Assume that we are designing a network S that can be represented by a simple graph $G = G(V, E)$ and the corresponding graph of cliques $G_{CL}(G) = G(V_{CL}, E_{CL})$, where $|V|$ is the number of vertexes of graph $G(S)$, and $|V_{CL}|$ is the number of cliques (possibly overlapping) in graph $G_{CL}(G)$. A special class of networks is represented by a tree directed graph that realizes the Boolean vector-functions $B_{k,m}$, i.e., k Boolean functions of m variables. Let $G(B_{k,m})$ be the graph of Boolean network $S(B_{k,m})$.

Assume that the graphs $G(S) \equiv G(V, E)$, $G_{CL}(G) \equiv G(V_{CL}, E_{CL})$ and $G(B_{k,m}) \equiv G(V_B, E_B)$ are characterized by the following parameters:²

- “general dimension” m ;
- number of vertexes $N = |V|$;
- global overlapping index (as a “clusterness” measure) $G_{CL}(G)$;
- diameter D of $G(S)$;
- algorithmic complexity $K(S(m))$;
- density graphs $G(S(m))$ and $G_{CL}(G(S(m)))$; and
- information cost $J(G(B_{k,m}))$.

Let $\Omega(m)$ be the following set of conditions (constraints)³:

1. $|V| = O(\exp m)$;
2. $|V_{CL}| = O(\exp m)$;
3. $K(S(m)) = O(\log m)$;
4. $\forall m (m \geq 3)$ density $G_{CL}(G(S(m))) = 1$;
5. $\forall m (m \geq 3)$ $D(G(S)) \leq 3$; and
6. $\forall k, m J(G(B_{m,k})) = O(\rho^{-1} m^4 k)$, where ρ is the probability of error for any of the edges.

Boolean networks were proposed by Kauffman (Kauffman 1969) to study gene regulatory networks (GRNs). A directed graph of GRNs is comprised of labeled nodes—the genes and their regulators. The inputs of these nodes are proteins (different transcription factors (TF)), and the outputs are the levels of gene expression. When modeling GRN by Boolean networks, estimating the information cost of computation in Boolean networks is especially desirable for detailed researching of the information flows. We will show that there is the network $S(B_{k,m})$ when computing the Boolean vector-functions $B_{k,m}$ for which constraint 6 holds true.

Aside from the Boolean networks, in this paper, we are studying a special class of labeled networks $S(m)$. Our main goal is the analysis of labeling methods that allow for the synthesis of graphs matching conditions 1-5 listed above. It opens the door to deterministic synthesis of networks with optimal robustness features. We develop labeling methods that may come close to meeting this goal. Numerical modeling shows that there

²All exact definitions for these parameters are given in the subsequent sections.

³As usual, $O(\dots)$ means “some function bounded above in absolute value by a constant times what is in the parentheses”.

is deterministic method for the synthesis of networks $G(S(m))$ for $m = 3, \dots, 9$ matching conditions $\Omega(m)$.

The paper is structured as follows. In Section 1, we introduce two functionals that are complexity measures for the Boolean network. The first, the so-called *entropy volume* of computation, quantifies the total “occupation” of all circuit channels under a given distribution of input values. The second one—the *information cost of computation*—evaluates the capacity of a circuit’s elements with respect to information transmission. For any $B_{n,m}$, one can design a network $S^*(B_{n,m})$, such that the upper bound for entropy volume and the information cost of computation for $S^*(B_{n,m})$ grows with n as $\rho \log^c(L(F_{n,m}))$, where $\rho > 0$ is a desired error of computation, $L(B_{n,m})$ is the complexity of $B_{n,m}$, and $c \leq 4$ is a constant.

In Section 2, we present an explicit construction of deterministic models of complex networks. In these models, the labels of nodes are different partitions of the integer m^2 ($m \geq 3$) into at most m integer parts, and connections between nodes are defined by the metric distance between partitions. The values of some topological parameters of graphs from different families are presented analytically, whereas the values of other parameters (size and number of cliques, diameter, average distance, energy and so on) can be computed numerically. A characteristic feature of these deterministic models is the simplicity of the network-generating algorithm, which nevertheless allows for the design of large and very attractive, as far as resilience is concerned, topological objects. It is not inconceivable that the predictability of the topological parameters of these networks is a consequence of their low K-complexity. Some results of numerical experiments on estimating the robustness of these partition networks (quantified by the proportion of nodes and cliques that can be removed before the network loses connectivity) were presented in (Goryashko et al. 2019).

It should be noted that the basic features of the deterministic model from this section were published in our paper (Goryashko et al. 2019).

Finally, Section 5 is focused on a discussion of the perspectives on the design of complex networks in the form of attributed graphs and connections of this approach to the current state of the art.

Informational complexity bounds for computing Boolean functions by combinational circuits

In analyses of complex networks, the most commonly used static complexity measures are the number of elements (nodes or edges of the graph) or the topological parameters of the network’s graph. However, with a focus on functional complexity, the values of such dynamical properties as, for example, the “number of elements in [an] activity state on each computing step”, can be no less important. For the semiconductor elements of logical networks, it can be naturally assumed that the element “activity” is connected with changing element output values. Similar problems were studied with regard to the theory of complexity beginning 60 years ago in Russia. Now, with the increased study of real-world complex networks (biological, technological, and social), it seems useful to discuss the definitions of a network’s activeness in the process of information transmission.⁴

⁴The results of Section 1 were established in (Goryashko and Nemirovski 1978); in view of the unavailability of (Goryashko and Nemirovski 1978) to Western readers (i.e., until 80 yh, the Journal had not been translated into English), we reproduce the original proofs below to make the paper self-contained.

Preliminary explanation

In the initial stages of very-large-scale integration (VLSI) design, the most important problem was to reduce the power dissipated by a circuit (see, e.g., (Yajima and Inagaki 1974)).

However, most of the proposed approaches were ad hoc and did not allow for a general understanding of limits on energy consumption when computing the arbitrary Boolean vector-functions. Moreover, in the optimal model (in terms of the number of logical-element synthesis methods), starting with the classic results of C. Shannon, the number of active elements is proportional to the total number of elements, that is, for most of the functions, it grows exponentially with the number of variables. In (Goryashko and Nemirovski 1978), for the first time, to the best of our knowledge, it was shown that the “energy cost of computation” is proportional to the information cost of computation rather than to the number of elements in the circuit. We believe that this observation is, in particular, useful when investigating real-life neural networks. A biological neuron, as is the case for most basic biological channels (axons), utilizes the principles implemented in (Goryashko and Nemirovski 1978)—after becoming active, they switch themselves off for refractory period, where no energy is consumed.

This, combined with appropriate changes in circuit synthesis as compared to the standard synthesis methods of that time, allows us to arrive at the results on energy consumption and reliability of computation stated in Theorems 1-4.

When an “active” element is defined as an element of logical network for which the output value or some of the input values changes, then every Boolean function can be implemented by a network with $O(n^2)$ active elements. It is unknown whether this estimate is unimprovable.

For the problem in which we are interested, every network S (combinational circuits) is characterized by two functionals. The first, the so-called “entropy volume of computation”, quantifies the total “occupation” of all network channels under a given distribution of input values. The second—the information cost of computation—evaluates the capacity of the circuit’s elements with respect to the information transmission.

Let network $S(f(x_1, \dots, x_n))$ represent combinational circuits from elements AND, OR, NOT, and during the computation in S input variables for x_1, \dots, x_n , and let the value of the boolean function $f(x_1, \dots, x_n)$ appear in the output at each instance of time, $t = 0, \tau, 2\tau, \dots, k\tau, \dots$

Let $\Psi(\mathbf{x}) = \Psi(x_1, \dots, x_n)$ be a distribution of input vectors $\mathbf{x}_1, \dots, \mathbf{x}_T$, and all input vectors are independent, with distributions $\Psi(\mathbf{x})$.

Let graph $G_S(V, E)$ be a directed graph, where $v \in V$ are elements of network S , and $e \in E$ are connections between the elements.

Assume that each vertex $v \in V$ has a discrete binary source without memory and that each edge $e \in E$ is a discrete binary noiseless channel. Each distribution $\Psi(\mathbf{x})$ in $S(f(x_1, \dots, x_n))$ can be used to set up the distribution Φ_s of binary symbols for each source s . For each $e \in E$, weight $H_{e, \Psi}$ can be assigned, where $H_{e, \Psi}$ is the entropy of the incident source when $\Psi(\mathbf{x})$ is the input distribution.

Definition 1. *The computation of entropy volume for the Boolean function $f(x_1, \dots, x_n)$ is defined as*

$$H(n, m, \Psi) = \inf_S \sum_{e \in E} H_{e, \Psi},$$

where the infimum is taken across all networks $S(f(n, m))$ in computing the Boolean vector-function $f(n, m) = f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)$.

Theorem 1. $\forall n, m \quad \forall f(n, m) \quad \forall \Psi(x) \quad \exists S(f(n, m)) : \left\{ H(f(n, m)) \leq c(n^3 + n^2 m) \right\} \& \left\{ |S(f(n, m))| \leq 3m \cdot 2^n \right\}$, where $|S|$ is the number of elements in network S , and c is a constant.

The proof is given by describing the construction of circuit S and verifying that this circuit indeed meets the theorem's conclusion. We will soon describe the construction only; the computation of $H(S(f(n, m)))$ and $|S|$ is straightforward.

The part of the network that can realize any set of different conjunctions of n variables is named the universal decoder (UD(n)). The network (UD(4)) is depicted in Fig. 1 and Fig. 2.

The network UD(4) is depicted in Fig. 2. Here, each rank j ($j = 0, \dots, n - 1$) consists of 2^j blocks, and each block realizes $2(n - j)$ conjunctive terms with $j + 1$ letters (input variables). For each block ξ ($\xi = (1, 2, \dots, 2^j)$), a conjunction in rank j is $(x_1^{\sigma_1}, \dots, x_j^{\sigma_j})x_\mu$, where $|\sigma_1, \dots, \sigma_j| = \xi$ and x_μ is taken as $x_{j+1}, \bar{x}_{j+1}, \dots, x_n, \bar{x}_n$. Rank 0 contains only n NO elements; the other ranks are built up from AND elements with two inputs. Each conjunction in rank $j + 1$ is formed from two conjunctions in rank j as $x_1^{\sigma_1} \dots x_j^{\sigma_j} x_{j+1}^{\sigma_{j+1}} x_{j+2}^{\sigma_{j+2}} = x_1^{\sigma_1} \dots x_j^{\sigma_j} x_{j+1}^{\sigma_{j+1}} \& x_1^{\sigma_1} \dots x_j^{\sigma_j} x_{j+2}^{\sigma_{j+2}}$. Therefore, the outputs of rank $n - 1$ are 2^n different conjunctions $(x_1^{\sigma_1}, \dots, x_n^{\sigma_n})$. Universal junction (UJ) is used to realize any function $f(n, m)$, made up of the OR elements, with two inputs.

Model for a channel with memory

Let $C(Q + 1)$ be a binary channel with a set of states $\mathbf{S} = S_0, S_1, \dots, S_Q$, and $S(t)$ is state of the channel in time step t . The state S_0 will be named as "active" and other states as "passive". The state transition graph is shown in Fig. 3. If the channel is active, its output equals its input; otherwise, its output is 0. After the transition $S_0 \rightarrow S_i$ for any $i = 1, \dots, Q$

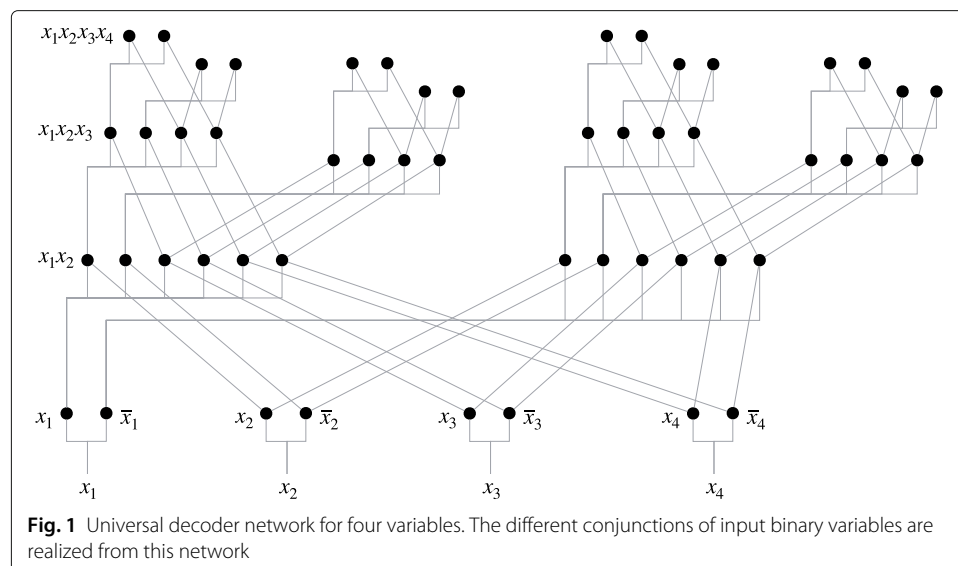
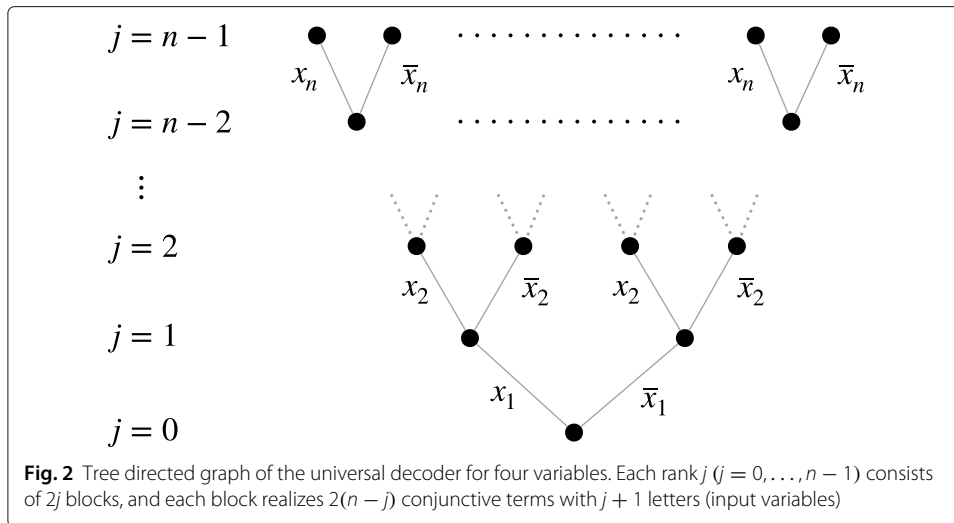


Fig. 1 Universal decoder network for four variables. The different conjunctions of input binary variables are realized from this network



the channel will be passive during the Q time step⁵. In a seminal work (Gallager 1970), general definitions of channel capacity with memory were provided as follows:

$$\begin{aligned} \underline{C} &= \lim_{T \rightarrow \infty} \underline{C}_T, \\ \bar{C} &= \lim_{T \rightarrow \infty} \bar{C}_T, \\ \underline{C}_T &= T^{-1} \max_{P(x^T)} \min_{S(t_0)} J_P(\Gamma(x^T); x^T), \\ \text{and } \bar{C}_T &= T^{-1} \max_{P(x^T)} \max_{S(t_0)} J_P(\Gamma(x^T); x^T). \end{aligned} \tag{1}$$

Here, $S(t_0)$ is the initial state of channel; $P(x^T)$ is the distribution of input words with length T ; $\Gamma(x^T)$ is the output words after transmission through Γ ; and $J_P(\Gamma(x^T); x^T)$ is the mutual information of random values x^T and $\Gamma(x^T)$, assuming that the distribution of input words is $P(x^T)$.

Theorem 2. For $C(Q + 1)$, the following statements are true:

1. $C(Q + 1) = C(Q + 1) \equiv C(Q + 1) = -\ln(1 - p^*)$, where p^* is the root of the equation $\ln p = (1 + Q) \ln(1 - p)$, and $p, (1 - p)$ are probabilities of 1 and 0 as channel outputs, accordingly;
2. $C(Q + 1) \leq [\ln(Q + 1) + 1] / (Q + 1)$.

The proofs of Theorem 2 and Theorems 3 and 4 are also in the Appendix.

Network $S_Q(f(n, m))$ in a Q -basis and the information cost of computation

Definition 2. The Q -basis (Q_B) has such logical elements AND, OR, and NOT such that for each logical element of Q_B ,

- Delay equals zero;
- The number of outputs equals r ($r \geq 1$); and
- Each output channel has memory equaling Q .

⁵Our model of a channel is similar to a physiological model of muscle cells or neurons, where there is such a condition as that called the refractory period.

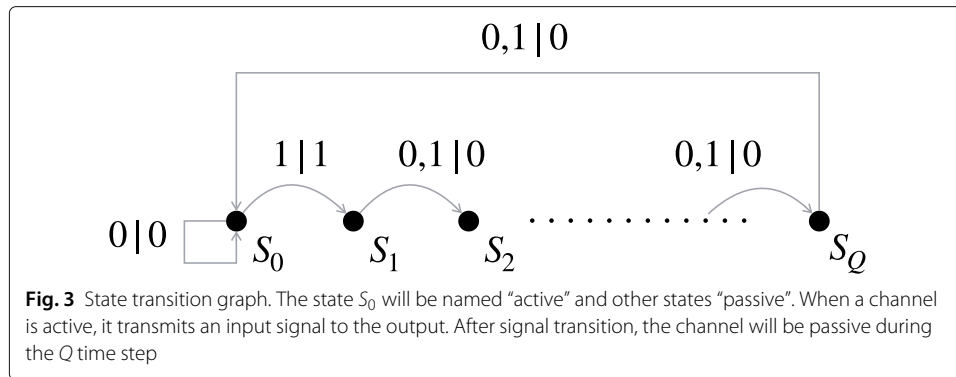


Fig. 3 State transition graph. The state S_0 will be named “active” and other states “passive”. When a channel is active, it transmits an input signal to the output. After signal transition, the channel will be passive during the Q time step

Definition 3. The capacity of element $e(j) \in Q_B$ is

$$C_j(Q) = r(e(j)) \cdot C(Q(e(j)))$$

, where $C(Q(e(j)))$ is the capacity of output channel $e(j)$.

Define the output of the ideal network $S_0(f(n, m))$, i.e., the network with $Q = 0$, with y_1, y_2, \dots, y_T , where each y_i consists of m bits of the $f(n, m)$ values. If for the network $S_Q(f(n, m))$, $Q > 0$, any channel $C(Q + 1)$ translates without error only the part with unit symbols. As a result of our network, $S_Q(f(n, m))$ will be *unreliable*, where output is z_1, z_2, \dots, z_T . Let it be reasoned that error exists if $y_i \neq z_i$. The total error during T computing steps is defined as $\rho(y^T, z^T)$. Let $\Psi(x) = \Psi(x_1, \dots, x_n)$ for the distribution of input vectors x_1, \dots, x_T , and μ_k^T is a random sequence of T signals in channel $k \in K$, and $H(T, k)$ is entropy of this sequence in the case of distribution $\Psi(x)$ for the input vector x^T .

Definition 4. The entropy volume of computation for network $S_Q(f(n, m))$ is

$$H(S, \Psi) = \lim_{T \rightarrow \infty} (1/T) \sum_{k \in K} H(T, k).$$

Definition 5. The information cost of network $S_Q(f(n, m))$ is

$$J(S_Q(f(n, m))) = \sum_{j \in S_Q(f(n, m))} C_j(Q_j).$$

Definition 6. The error of the computing function $f(n, m)$ by the network $S_Q(f(n, m))$ at time T is the quantity

$$\rho(S_Q(f(n, m)), T) = (1/T) \sum_{x^T} \rho(y^T, z^T) \Psi^T(x^T).$$

(Note that the definition of this quantity is nothing but the definition of error probability in a sequence of T symbols (Gallager 1970).)

The error of computing $f(n, m)$ by network $S_Q(f(n, m))$ is defined as

$$\rho(S_Q(f(n, m)), T) = \lim_{T \rightarrow \infty} \rho_\Psi(S_Q(f(n, m)), T).$$

Definition 7. The information cost of the computing function $f(n, m)$ with error ρ is defined as

$$J(n, m, \rho) = \inf_{S_\rho} J(S(f(n, m))),$$

where the infimum is taken across all networks S_ρ , computing $f(n, m)$ with an error that, at most, is ρ .

In (Goryashko and Nemirovski 1978), it was proven that given a function $f(n, m)$ and any distribution of inputs $\Psi(\mathbf{x})$, it is possible to select the values of Q for all elements of the network described in Theorem 1 to ensure the validity of the following statement.

Theorem 3. $\forall n, m (n \geq m)$, there exists a network $S_\rho (f(n, m))$ that computes $f(n, m)$ with error not exceeding ρ such that

$$\{J(S_\rho (f(n, m))) \leq 4\rho^{-1}n^4m\} \& \{|S_\rho (f(n, m))| \leq 3m \cdot 2^n\}.$$

In (Goryashko and Nemirovski 1978), it was also shown that the method of synthesis from Theorem 3 provides the following upper bound of informational cost: if the error of computing $f(n, m)$ is $\rho + \sum_{i=1}^m \rho_i$, then $J(S_\rho) \leq \rho^{-1}n^4 + n^3 \sum_{i=1}^m (\rho_i)^{-1}$.

Theorem 4. $\forall n, m(m \leq n) \forall \Psi(\tilde{x}) \forall f_{n,m} \forall \rho(\rho > 0) \exists (\mathfrak{A}(f_{n,m})$ computing $f_{n,m}$ with error $\leq \rho$): $\{J(\mathfrak{A}(f_{n,m})) \leq 4\rho^{-1}n^4m\} \& \{|\mathfrak{A}(f_{n,m})| \leq 3m2^n\}$.

Graphs labeling by partition: basic definitions

Our basic goal is to invent a method for the synthesis of simple graphs that possess the following properties:

- low algorithmic complexity;
- the ability to create rich, evolving topological structures; and
- low time consumption for the numerical simulation of the resulting structure.

It turns out that all these properties can be achieved by a special method of attributing graph nodes, which in (Goryashko et al. 2019) was named *partition family graphs*. Here, we restate the basic definitions.

Definition 8. A (n, m) -partition of n into no more than m parts is defined as a sequence of nonnegative integers $a_1 \geq a_2 \geq \dots \geq a_m \geq 0$, such that $n = a_1 + a_2 + \dots + a_m$. The set of all feasible (n, m) -partitions is denoted by $P(n, m)$ ⁶.

Definition 9. For (n, m) -partition $\alpha = (a_1, \dots, a_m)$, the gaps between parts are the values $k_i = a_i - a_{i+1}$ ($1 \leq i \leq m - 1$). A partition whose gaps k_i are the same as and equal to some specific $k \geq 2$ is called a partition with uniform gaps, and the set of all these partitions is denoted by $UGP_k(m)$. Note that any partition set $P(m^2, m)$ where $m \geq 3$ contains $UGP_2(m)$, which is the sequence of odd numbers $(2m - 1, 2m - 3, \dots, 3, 1)$.

Definition 10. The distance between partitions $\alpha = (a_1, \dots, a_m)$ and $\beta = (b_1, \dots, b_m)$ ($\alpha, \beta \in P(n, m)$) is defined as

$$\rho(\alpha, \beta) = \max_{i=1, \dots, m} |a_i - b_i|. \tag{2}$$

For all $\alpha, \beta \in P(n, m)$ ($\alpha \neq \beta$), it holds that $2 \leq \rho(\alpha, \beta) \leq 2n(1 - 1/m)$. The least upper bound is achieved for $\alpha = (n, 0, \dots, 0)$ and $\beta = (n/m, n/m, \dots, n/m)$.

Definition 11. Let a graph $G(V_{\mathcal{P}}, E)$ consist of a set of vertices (nodes) V , where each node has unique label $\alpha \in \mathcal{P} \subseteq P(n, m)$, and every two nodes $v_\alpha, v_\beta \in V_{\mathcal{P}}$ are connected by an edge $e \in E$ iff $\rho(\alpha, \beta) = 1$.

What is the direct descendant partition family?

In (Goryashko et al. 2019), the so-called trinomial partition family $TPF(m)$ was introduced as the set of partitions $\alpha = (a_1, \dots, a_m)$ with the head partition $h = (2m - 1, 2m -$

⁶There exist an optimal algorithm and recurrent rules for the exact computation of the number of partitions in the classes $P(n, m)$ (Knuth 2011).

3, . . . , 1) and for which $\rho(\alpha, h) = 1$, i.e., any partition $\alpha \in \text{TPF}(m)$ is adjacent to a head (with a head being the hub of a graph for $\text{TPF}(m)$). Therefore, for any two nodes α, h , the sequence $\mathfrak{R}_m = (\delta_1, \delta_2, \dots, \delta_m)$, where $\delta_i = h_i - a_i$, is ternary sequence permitting the values (1, 0, -1) and $\sum \delta_i = 0$. It is known (Andrews 1971; 1990) that the number of permutations \mathfrak{R}_m exactly equals the values of the trinomial triangle (generalization of Pascal’s Triangle) depicted in Fig. 4.

Following the notation (Andrews 1971), the trinomial coefficient $\binom{m}{k}_2$ with $m \geq 0$ and $-m \leq k \leq m$ is given by the coefficient of x^{m+k} in the expansion of $(1 + x + x^2)^m$.

The central trinomial coefficient ($k = 0$) for $m = 3, 4, 5, 6, 7, 8, 9, 10, \dots$ is given by the sequence 7, 19, 51, 141, 393, 1107, 3139, 8953, . . . (sequence A002426 in the OEIS (OEIS Foundation Inc 2018)). If $k = 1, k = -1$ trinomial coefficients is given by the sequence 6, 16, 45, 126, 357, 1016, 2907, 8350, . . . (sequence A005717 in the OEIS). The central trinomial coefficient is asymptotic: $a(m) \approx d \cdot 3^m / \sqrt{m}$, where $d = \sqrt{3/\pi} / 2$ (OEIS Foundation Inc 2018).

Now, we will examine the extended class of partitions which will be named as the direct descendant partition family (DDPF(n, m)), which contains partitions with m parts but with different values of n (“spectrum” of values $n = m^2 \pm l, l = 0, 1, 2, \dots, m - 2; m \geq 2$).

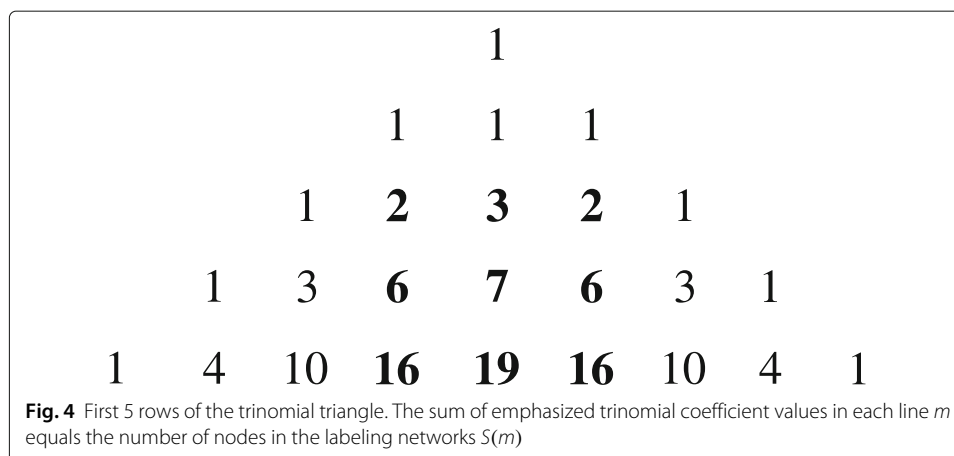
For a partition set DDPF(n, m), subset DDPF(m^2)—i.e., when $l = 0$ —will be named a progenitor for any family from DDPF(n, m) because any set of partitions $F(m, n) \in \text{DDPF}(n, m)$ is created easily from the subset $P(m^2) \equiv P$.

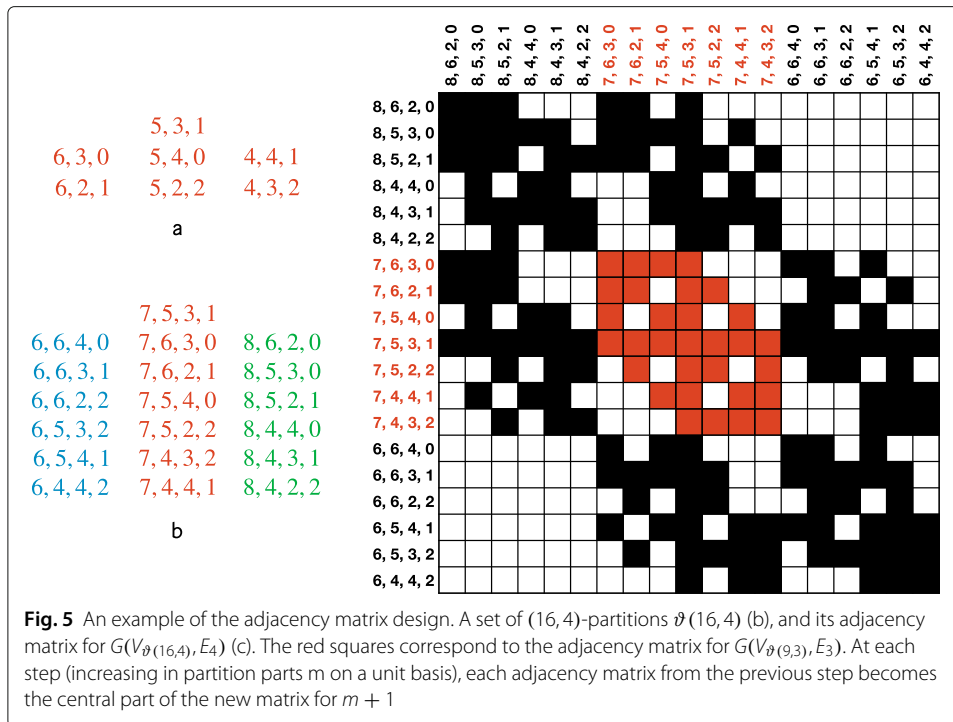
Thus, we have the possibility of estimating a priori the number of adjacent partitions for progenitor. More importantly, it turns out that there is a simple recurrent procedure for designing the graphs $G(V_P, E)$ and, consequently, the graph $G(V_{F(m,n)}, E)$.

Recurrent procedure for designing the graph of $G(V_P, E)$

The procedure under examination is founded on the special labeling of graph nodes by trinomial partitions and, due to such labeling on the design of the graph with partitions, $P(m^2)$ from the graph with partitions $P((m - 1)^2)$ using simple arithmetical operations.

- *Step 0.* Let us have the initial partition $P(9, 3)$ (Fig. 5a).
- *Step 1.* First, to create partitions in accordance with the central trinomial coefficient for $P(16, 4)$, we prepend the odd number $a_1 = 2m - 1$ to all partitions from *Step 0* (Fig. 5b, red color).





- **Step 2.** To find two side subsets ($k = 1, k = -1$) of trinomial coefficients (see Fig. 5b, green and blue color accordingly) it is sufficient to define all partitions such that the following holds:
 - for all partitions, $n = m^2$;
 - the first part of all partitions equals $2m$ when $k = 1$ (green column);
 - first part of all partitions equals $2m - 2$ when $k = -1$ (blue column); and
 - for all partitions, $\alpha \rho(\alpha, h) = 1$.
- **Step 3.** To design the graph $G(V_{P(m^2, m)}, E_m)$, we have to connect any nodes α, β iff $\rho(\text{label } \alpha, \text{label } \beta) = 1$.

In Fig. 5c, the adjacent matrix of $G(V_{P(16,4)}, E_4)$ is depicted.

Let us estimate the designing cost T_m of the graph $G(V_{P(m^2, m)}, E_m)$ as an upper bound on its running time. Each partition $\alpha \in P(m^2, m)$ can be presented by word $w(\alpha)$ from $2 \log_2 m$ binary digits. To find all partitions adjacent to the partition α , it is sufficient to make $3m$ arithmetical operations under the numbers $w(\alpha)$. The total number of computational steps number no more than $3 \cdot 3 \cdot m \cdot L(m)$ arithmetical operations, where $L(m)$ is the number of partitions for the central trinomial coefficient. Note that the number of nodes in $G(V_{P(m^2, m)}, E_m)$ is no greater than $3 \cdot L(m)$.

At the same time, it is known that for random graphs, the expected runtime is $O(N+M)$, where N is the number of nodes, and M is the expected number of edges (Miller and Hagberg 2011).

Except for the running time $T(m)$, there is another measure cost: the algorithmic complexity or K -complexity (Kolmogorov 1965). K -complexity (s) is defined for the strings of characters s as

$$(s) = \min\{|P|, T(P) = s\}, \tag{3}$$

where P is a program that produces the string s when executed on the universal Turing machine, and $|P|$ is the length of the program P —the number of bits required to represent P . The K -complexity is upper semicomputable, i.e., only the upper bound of the value of K -complexity can be computed for a given string s . As is easy to see from description of this procedure, the upper bound of the number of bits required to represent the adjacency matrix of $G(V_{\vartheta(m^2, m)}, E_m)$ is $O(\log m)$ because only the value of m is necessary to create the appropriate program.

As is clear from the description of the recurrent procedure and Fig. 5c at each step (increasing the partition parts m on unit basis), the adjacency matrix from the previous step becomes the central part of the new matrix for $m + 1$. This principle of iterative nesting provides for the effect of self-similarity (similar to a Russian matryoshka doll).

The detailed study of topological parameters in real-world objects—most importantly, biological objects—is of prime interest for possible application of the partition families. (Strang et al. 2018) shows the results of an analysis of 128 graphs designed from the experimental results of studying brain communities. It turns out that numerical values of global efficiency, density and characteristic path length are very close to the values that were calculated for progenitor DDPF(m^2) graphs when $m = 6, 7$, and connections between these parameters coincides with the findings of the authors. For example, in (Strang et al. 2018), for average values, $E_{glob}^*(G) = 0.5(1 + Dens(G))$, and $L^*(G) = (2 - Dens(G))$. From Table 1 for DDPF(m^2) and $m = 6$, the value of $L(G_6) = 1.7$, and $Dens(G_6) = 0.28$. The estimate $E_{glob}^*(G) = 0.64$, and $L^*(G) = (2 - 0.28) = 1.72$. Differences between values from Table 1 and average estimates from (Strang et al. 2018) increase as m increases, but methods for measuring topological parameters for the brain are inexact, especially when they relate to the number of elements and their density. In any case, such coincidences invite further investigation.

Table 1 The results of numerical calculation of the graph’s connectivity (parameters $MDenV(m)$ and $MDenCl$ are calculated for graphs **without head nodes**)

m	$ V $	Node	$D(1)$	$D(2)$	$D(3)$	$D(4)$	$D(5)$	Md	$MdenG$	$MdenCl$	Ov	L	Cc
3	7	Deg.	3	6	–	–	–	3.4	0.87	1.0	2.57	1.43	0.54
		Quant.	6	1	–	–	–						
4	19	Deg.	5	9	18	–	–	8.2	0.39	1.0	2.58	1.54	0.59
		Quant.	6	12	1	–	–						
5	51	Deg.	13	25	50	–	–	18.2	0.34	1.0	4.9	1.63	0.56
		Quant.	30	20	1	–	–						
6	141	Deg.	19	35	69	140	–	40.3	0.28	1.0	6.53	1.7	0.52
		Quant.	20	90	30	1	–						
7	393	Deg.	49	95	191	393	–	87.9	0.22	1.0	8.72	1.77	0.47
		Quant.	140	210	42	1	–						
8	1107	Deg.	69	132	261	533	1106	194.7	0.17	1.0	11.6	1.82	0.42
		Quant.	70	553	420	55	1						
9	3139	Deg.	181	357	752	1499	3138	437.4	0.14	1.0	15.5	1.86	0.37
		Quant.	630	1681	755	74	1						

The following notation is used: $D(j)$ is the index of specific vertexes degree, $Cc(m)$ is the clustering coefficient, $Md(m)$ is the average node degree, $MDenCl(m)$ is the density of clique graph $G_C(V, E)$, Ov is the global overlapping index, and L is the characteristic path length.

Topological parameters of the progenitor partition family $DDPF(m^2)$ graphs

Let us introduce some often-used topological parameters of graphs and provide reminders of definitions (see details in (Strang et al. 2018)) (the programming tools for computer estimates of these parameters can be found in (Samokhine 2017)). Beyond this point, we assume that $G(V, E)$ is a simple graph with $|V| \equiv N$ vertices and $|E| \equiv M$ edges. The *graph density* $den(G) = 2M/N(N - 1)$ and $d(i, j)$ represent the *distance* (the number of edges in the shortest path between vertices i and j in $G(V, E)$). If there is no path connecting i and j , then $d(i, j) = \infty$. The *diameter* $D(G(V, E)) = \max_{ij} d(i, j)$. The *characteristic path length* L is the average distance over all pairs of vertices, i.e., $L = 1/(N(N - 1)) \sum_{i \neq j} d(i, j)$. In (Strang et al. 2018), a measure named the *global efficiency* was proposed: $E_{glob}(G) = 1/(N(N - 1)) \sum_{i \neq j} (1/d(i, j))$. The *local efficiency* is the average of the global efficiencies of all subgraphs G_i , i.e. $E_{loc}(G) = (1/N) \sum_{i \in G} E_{glob}(G_i)$. Beyond the common parameters listed above, we are in need of some measure for estimating of overlapping nodes. Let each node $i \in V$ from graph $G(V, E)$ be incorporated into the $\phi(i)$ cliques such that $0 \leq \phi(i) \leq$ total cliques of $G(V, E)$. The value $\phi(i)$ will named the *local overlapping*. The global overlapping is the average of the local overlapping $Ov(G) = (1/N) \sum_{i \in G} \phi(i)$.

Overlapping problems

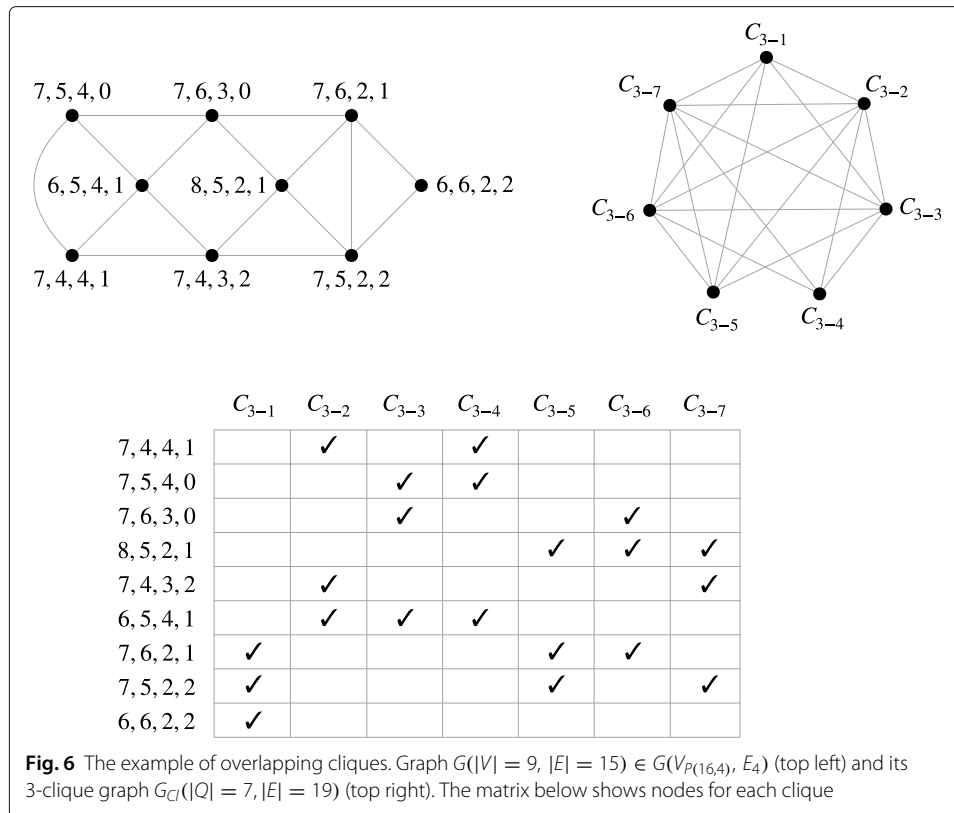
In recent years, interest in clique graphs has grown (see (Evans 2010; Fortunato 2010; Leskovec et al. 2010) and references herein). A large body of work has been dedicated to community detection methods and identification of overlapping communities in large real-world networks such as social networks, biochemical networks, and scientific publications as well as collaboration networks.

Usually, the reasons for the emergence of graph overlapping communities in networks did not attract the particular interest of scientists, although (Crandall et al. 2008) analyzed overlapping community models. There, some mechanisms were established by which the communities in LiveJournal and DBLP networks were growing and changing. In particular, it was shown that networks' structural changes directly depend on the quantity of overlapping clusters.

In the case under study, the numbers and sizes of the cliques' overlapping depends on nodes label only. By changing the nodes' labeling, one can gain insight into the emergent topological peculiarities of overlapping communities.

In the following, we shall deal also with the graph of cliques $G_{Cl}(Cl, E_{Cl})$, which is derivative of a simple graph $G(V_{P(m^2, m)}, E_m) \equiv G$. The nodes of $G_{Cl}(Cl, E_{Cl})$ are all cliques (size ≥ 3) of G , and set E_{Cl} is the edges set between all cliques. The example in Fig. 6 shows subgraph $G(|V| = 10, |E| = 19) \in G(V_{P(16, 4)}, E_4)$ and its clique graph $G_{Cl}(|Cl| = 6, |E_{Cl}| = 14)$ with global overlapping $Ov(G_{Cl}) = 2.2$. The density of G equals 0.42, and the density of $G_{Cl} = 0.9$. Therefore, in this case, the graph cliques are "almost" one 6-clique. Indeed, although the diameter of graph G is 3 for G_{Cl} , the characteristic path length is 1.71, and G_{Cl} has fewer than half the number of edges of G .

It may appear that this approach using a clique graph is no better than a trick because the total numbers of nodes and edges in G and G_{Cl} do not change. But in such cases when the cliques may be thought of as a unified entity (for example, from a point of functionality), such an approach has the potential for yielding information about optimal partition methods of large objects.



For example, one can look at the creation of some large scientific collaboration. Let the network nodes be scientists and the edges be the connections between the scientists. The success of the collaboration depends on—to a greater extent than the creative skills of each one person—the reconcilability (tolerance) of coworkers. If there are some groups of scientists (“cliques”) that have demonstrated successful interactions in the past, it makes sense to leave these cliques and create the network as clique graph.

Computer modules of some brain regions (gyrs; see (McCarthy et al. 2014)) have been studied as clique graphs (Strang et al. 2018) as well. Hence, for real-world biological, technological and computer networks, the properties of the overlapping communities could be governing factors, with point representing resilience, effectiveness or density.

Table 2 shows the distributions of the cliques of the progenitor $DDPF(m^2)$ that were established by numerical modeling for $m = 3, 4, \dots, 9$. The main peculiarity of the data from Table 2 is their predictability. It turns out that for $m = 3, 4, \dots, 9$, clique sizes are given by the first $\lfloor m/2 \rfloor$ binomial coefficients of Pascal’s triangle, and the total number of cliques equals $2m - 2$. There is good reason to believe that this property is true for any value m , but it is merely a hypothesis at present.

As one can see from Table 1, the distribution of the degrees of vertices resembles a binomial distribution (distinct from the power law for random graphs). Naturally, the diameter of the progenitor graph $G(DDPF(m^2))$ is different from that of a randomly evolving network as well. For example, the asymptotic estimate of the diameter D for the Balabos-Riordan network model is $D(|V|) \equiv \ln|V| / \ln \ln|V|$ (Bollobás and Riordan 2004). Evidently, for the progenitor graph $G(DDPF(m^2))$ with head node for any m , the diameter equals 2. It can be proven that if the head node were excluded from the progenitor

Table 2 The results of numerical calculation of the clique parameters (size and quantity)

m	$ V $	Cliques in $G'(m)$								Total
		$\binom{m}{1}^*$	$\binom{m}{2}$	$\binom{m}{3}$	$\binom{m}{4}$	$\binom{m}{5}$	$\binom{m}{6}$	$\binom{m}{7}$	$\binom{m}{8}$	
3	7	3_3	3_3	–	–	–	–	–	–	6
4	19	4_4	6_6	4_4	–	–	–	–	–	14
5	51	5_5	10_{10}	10_{10}	5_5	–	–	–	–	30
6	141	6_6	15_{15}	20_{20}	15_{15}	6_6	–	–	–	62
7	393	7_7	21_{21}	35_{35}	35_{35}	21_{21}	7_7	–	–	156
8	1107	8_8	28_{28}	56_{56}	70_{70}	56_{56}	28_{28}	8_8	–	254
9	3139	9_9	36_{36}	84_{84}	126_{126}	126_{126}	84_{84}	36_{36}	9_9	510

* $\binom{m}{i}$ —binomial coefficient ($i = 1, \dots, 8$).

graph, then for any m , $D(G(DDPF(m^2))) = 3$. As shown in Table 1, the characteristic path length remains less than 2 until $m = 9$, until the total nodes of graph number greater than 3000.

More surprising is the behavior of the average index density for the cliques graph. If the value of the average index density of the graph $G(DDPF(m^2))$ decreases with m fairly slowly (approximately as $1/\ln|V|$ for $m = 4, \dots, 9$), and if this represents more-or-less typical behavior, the cliques progenitor graph for any m will have a density equaling 1. This effect is reminiscent of the “densification power law” (Leskovec et al. 2010), which implies the existence of evolving networks for which the density tends to increase. In our case, the reason behind the “stable maximal density” of the clique graph is its high value of global overlapping.

Topological parameters of the descendant partition family (DDPF(n, m))

The topological properties of the progenitor networks $G(DDPF(m^2))$ holds for $m - 2$ different networks only. It turns out that the family $DDPF(n, m)$ with the “spectrum” of values $n = m^2 \pm l(m)$ possesses many topological properties of The family $DDPF(m^2)$ (Goryashko et al. 2019). Experimental results for $0 \leq l(m) \leq 4$ are partly shown in Table 3. Although the number of nodes for these graphs $G(DDPF(m^2 \pm l(m)))$ does not coincide with the graphs $G(DDPF(m^2))$, the total clique numbers are the almost the same (2^m versus $2^m - 2$). Their topological characteristics, such as the cluster coefficient, global efficiency, characteristic path length and graph density (with the only exception being the degrees of nodes distribution) for the networks in Table 3 are practically indistinguishable from the respective characteristics presented in Table 1.⁷

Experimental observation. In accordance with the experimental results, it could be supposed that distributions of clique numbers and sizes are defined by the m and ratio n/m values only. This would explain why the parameters are highly predictable.

Concluding remarks

The design of networks with attributes described by partitions is a first step on the way to a theory of evolving networks with prearranged topological properties. There are many open problems that call for further investigation before this approach could become a working instrument.

⁷The numerical values of almost all topological parameters for case of $(m^2 \pm l, m)$ -partitions is found to be very close to previous case (different by less than one percent).

Table 3 The partition family DDPF(n, m) for $n = m^2 \pm l(m)$, where $l(m) = 0, 1, \dots, 4$

m	$n = m^2$		$n = m^2 \pm 1$		$n = m^2 \pm 2$		$n = m^2 \pm 3$		$n = m^2 \pm 4$	
	Total nodes	Global overlap	Total nodes	Global overlap	Total nodes	Global overlap	Total nodes	Global overlap	Total nodes	Global overlap
2	—	0	7	0	9	0	9	0	—	—
3	7	2.57	19	2.63	13	2.46	12	2.0	—	—
4	19	3.57	51	3.56	39	3.23	27	3.18	—	—
5	51	4.90	141	4.77	111	4.43	81	4.22	50	4.0
6	141	6.53	393	6.38	321	5.96	241	5.65	182	5.45
7	393	8.72	1107	8.52	924	8.04	714	7.6	546	7.38
8	1107	11.62	3138	11.39	2674	10.79	2114	10.21	1646	9.92

As in Table 1 the parameters $MDenV(m)$ and $MDenCl$ are calculated for graphs without head node. Number of nodes for the graphs G (DDPF(n, m)) and Global overlapping does not coincidence with graphs for progenitor DDPF(m^2). Total cliques numbers are the almost the same (2^m instead of $2^m - 2$) and the topological characteristics such as clustering coefficients, global efficiency, characteristic path length and graph density (exception is the nodes degrees distribution) for networks of Table 3 are practically indistinguishable from accordance characteristics in Table 1.

One of the most intriguing problems is robustness, both in technical networks (for example, computing networks) and biological networks (for example, transcriptional regulatory networks (TRN)); for more details, see (Whitacre 2012) and references therein. It is well known that methods providing high robustness that were “invented” through billions of years of evolution of any living entity from fungi to mammals are fundamentally different from today’s engineering methods. However, studying the problem of robustness in the complex networks field as a rule takes its origin in the classical models of logical networks. As a result, it is difficult to evaluate the advantages of robust topologies in complex regulatory networks achieved through neutral evolution. We speculate that graph models where labels of each node contain information about specifics and connectivity of the node (something akin to gene sequences) will be more appropriate for modeling TRN, with such models including complex regulatory network topologies.

The partition labeling of network nodes makes it possible to develop a fresh approach to behavioral game theory, in particular, to Colonel Blotto and Lotto games (see (Bocharov and Goryashko 2017)). The results of numerical experiments show that there is tight connection between equilibria in mixed strategies in antagonistic Blotto-type games and topological parameters of networks with nodes labeled by game strategies.

From the viewpoint of practical applications, it is interesting to study network models where nodes are attributed by economical parameters. For example, each country could be represented by a network node labeled by a partition with m parts presenting the percentages of total expenditures going to a specific area—e.g., defense, health, and education; this maps onto the standard COFOG (Classification of the Function of Government) data. Topological parameters of such a network with N nodes (for N countries) would provide insight into some special features of international behavior.

We hope that interesting results can eventually be found in this field.

Appendix

Theorem 2. For the channel with memory Q described above, the following takes place:

1. $\underline{C}(Q) - \overline{C}(Q) \equiv C(Q) = -\ln(1 - p^*)$, where p^* is the root of the equation $\ln p = (1 + Q) \ln(1 - p)$; p and $1 - p$ are, respectively, the probabilities of having 0 and 1 as the channel’s input; and

$$2. \quad C(Q) \leq [\ln(Q + 1) + 1] / (Q + 1).$$

Theorem 3. 1. $\forall n, m \forall f_{n,m} \forall \rho (\rho > 0)$

$$J(n, m, \rho) \geq H_\Psi(y)m - [\rho m + H(\rho)].$$

2. $\forall n, m \forall f_{n,m} \forall \rho (\rho > 0)$ for every circuit $\mathfrak{A}^R(f_{n,m})$ computing $f_{n,m}$ with error $\leq \rho$, it holds that

$$J(\mathfrak{A}^R(f_{n,m})) \geq R\{H_\Psi(y)m - [\rho m + H(\rho)]\}.$$

Here, $H_\Psi(y)$ is entropy per symbol of the output alphabet of circuit \mathfrak{A} , Ψ is the input distribution, and $H(\rho) = \rho \ln(\rho) + (1 - \rho) \ln(1 - \rho)$.

Proof. Let the circuit \mathfrak{A} be characterized by the collection of transformations $\Gamma_T : A_n^T \rightarrow A_m^T$ and the circuit $\tilde{\mathfrak{A}}$ by the collection of transformations $\Gamma_T^0 : A_n^T \rightarrow A_m^T$. Let $J_\Psi(\Gamma_T(x^T); \Gamma_T^0(x^T))$ be the mutual information of random variables $\Gamma_T x^T, \Gamma_T^0 x^T$ corresponding to the input distribution $\Psi(x^T)$. Then,

$$J_\Psi(\Gamma_T x^T; \Gamma_T^0 x^T) \leq J_\Psi(x^T; \Gamma_T x^T) \leq \max_\Psi J_\Psi(x^T; \Gamma_T x^T) \equiv TC_T(\mathfrak{A}).$$

By definition, $C(\mathfrak{A}) \geq \overline{\lim}_{T \rightarrow \infty} C_T(\mathfrak{A})$, where $C(\mathfrak{A})$ is the channel capacity of circuit \mathfrak{A} treated as a communication channel.

From the definition of circuits (Section 4, item 3) and information cost of computation (Section 4, item 7), it is easily seen that $J(\mathfrak{A}(f_{n,m})) \geq C(\mathfrak{A}(f_{n,m}))$ and $J(\mathfrak{A}^R(f_{n,m})) \geq RC(\mathfrak{A}^R \dashrightarrow (f_{n,m}))$ (with the latter inequality following from the fact that the total amount of information passing from level to level cannot be less than the total amount of information passing from the input to the output of \mathfrak{A}^R).

Thus, we conclude that if $\mathfrak{A}(\mathfrak{A}^R)$ computes $f_{n,m}$ with error $\leq \rho$, then

1. $J(\mathfrak{A}(f_{n,m})) \geq \overline{\lim}_{T \rightarrow \infty} T^{-1} J_\Psi(\Gamma_T x^T; \Gamma_T^0 x^T) \equiv J_\Psi$; and
2. $J(\mathfrak{A}^R(f_{n,m})) \geq RJ_\Psi$.

From this assumption regarding $\mathfrak{A}(\mathfrak{A}^R)$, $\overline{\lim}_{T \rightarrow \infty} T^{-1} \sum_{x^T} (\Gamma_T x^T, \Gamma_T^0 x^T) \Psi^T(x^T) \leq \rho_*$.

According to the inequality obtained when inverting the Coding Theorem ([6], p. 98), it follows that $J_\Psi \geq mH_\Psi(y) - [\rho m + H(\rho)]$. The theorem is proven.

Theorem 4. $\forall n, m (m \leq n) \forall \Psi(\tilde{x}) \forall f_{n,m} \forall \rho (\rho > 0) \exists (\mathfrak{A}(f_{n,m})$ computing $f_{n,m}$ with error $\leq \rho$): $\{J(\mathfrak{A}(f_{n,m})) \leq 4\rho^{-1}n^4m\} \& \{|\mathfrak{A}(f_{n,m})| \leq 3m2^n\}$.

Proof. Let us point out the rules for selecting memory sizes for the Q -elements of the UD circuit. Let us make the memory sizes for all elements of a block ξ_j of level j equal to $G(\xi_j)$. The information cost J_{UD} of the decoder with a collection of memories $Q = \{Q(\xi_j), j = 0, 1, \dots, n - 1\}$ clearly satisfies the bound

$$(2) \quad J_{UD}(Q) \leq \sum_{j=0}^{n-1} 2(n-j) \sum_{\xi_j \in A-j} C(Q(\xi_j)),$$

where A_j is the set of binary words of length j . Let us upper-bound the probability of decoding the error at a given time t . Note that with our construction, errors of only one type are possible, namely, the absence of signals at all 2^n output nodes; that is, "erroneous" output produces a signal. Now, let a vector \tilde{x} input arrive at time t . It will be decoded with error only when at least at one of the levels j the elements of block $x^{(j)}$ at time t are in the non-working state (here $x^{(j)} \in A_j$ is the vector of the first j symbols in vector \tilde{x}). In turn, the elements of the block $x^{(j)}$ can be in the nonworking state only when for at least

one of the time instants $t - 1, \dots, t - Q(x^{(j)})$, the vector \hat{x} in the decoder's input was such that $\hat{x}^{(j)} = \tilde{x}^{(j)}$. The conditional probability of the latter event, the condition being that the input at time t is \tilde{x} , is not greater than $Q(x^{(j)})p_j \cdot (x^{(j)})$, where $p_j(\xi)$, $\xi \in A_j$, if the distribution of the first j input symbols is induced by the distribution Ψ .

Consequently, the probability $p(\text{err}|x)$ of error at time t , the input being x , is no greater than $\sum_{j=0}^{n-1} Q(x^{(j)})p_j(x^{(j)})$, and the unconditional probability $p_{\text{err}}(t)$ of error at time t satisfies the bound

$$(3) \quad p_{\text{err}}(t) \leq \sum_{x \in A_n} \left\{ \sum_{j=0}^{n-1} Q(x^{(j)})p_j(x^{(j)}) \right\} p(x).$$

By first summing over x and then over j , after straightforward computations, (3) can be rewritten as

$$(4) \quad p_{\text{err}}(t) \leq \sum_{j=0}^{n-1} \sum_{\xi_j \in A_j} Q(\xi_j)p_j^2(\xi_j).$$

Now, let us minimize the right-hand side in (2) under the restriction that the right-hand side in (4) is no greater than ρ and utilizing in (2) the bound for $C(Q)$ from Theorem 2 instead of $C(Q)$. From the Lagrange rule, we posit a good guess as

$$(5) \quad Q(\xi_j) = \lambda \sqrt{j(n-j)} / p_j(\xi_j).$$

With $\lambda = 2\rho n^{-2}$ and

$$(6) \quad Q^*(\xi_j) = [2\rho \sqrt{j(n-j)} / n^2 p_j(\xi_j)],$$

the right-hand side in (4), and thus, the probability of decoding error at (any) time t is no greater than ρ . Thus, selecting memory according to (6) ensures the required error level. From (2), we easily obtain the following bound on the information cost of the decoder yielded by (6):

$$(7) \quad J_{UD}(Q^*) \leq \rho^{-1} n^4.$$

Let us denote by \mathfrak{M} the decoder just described, and let us select the memory sizes in the assemblies \mathfrak{M}_i corresponding to the components $f_{n,i}$ of the vector-function $f_{n,m}$. Let $B = \{b\}$ be the set of outputs of the decoder \mathfrak{M} . Let us fix $i \in \{1, \dots, m\}$ and focus on assembling \mathfrak{M}_i . Let $U_0 \in B$, $U_1, \dots, U_s =] \log |U_0| [$ be the set of nodes of the assembly of levels $0, \dots, s$, respectively. Note that $|U_j| \leq 2^{s-j}$. For node $\eta \in U_j$, let $U_j(\eta)$ be those linked to the η nodes from the set U_0 . Denoting the memory of a node $\eta \in U_j$, $0 \leq j \leq s - 1$, by $Q(\eta)$, the information cost of assembly \mathfrak{M}_i satisfies the bound

$$(8) \quad J(\mathfrak{M}_i, Q(\eta)) \leq 2 \sum_{j=0}^{s-1} \sum_{\eta \in U_j} C(Q(\eta)).$$

To ensure reliability of the assembly, we act as follows. Let $\tilde{\mathfrak{M}}$ be a decoder with no memory of the same form as \mathfrak{M} , and let $p(b)$ be the probability of getting a signal on the output $b \in B$ of the decoder $\tilde{\mathfrak{M}}$ when the distribution of inputs is Ψ . Let us compare two circuits: \mathfrak{A}_i comprised of a serial arrangement of decoded \mathfrak{M} and assembly \mathfrak{M}_i and similar combination $\tilde{\mathfrak{A}}_i$ of circuits $\tilde{\mathfrak{M}}$ and $\tilde{\mathfrak{M}}_i$; moreover, let them work on the same sequence of inputs x^T . Since the only possible error for \mathfrak{M} is the absence of any output signal at all, from the construction of \mathfrak{M} , it is clear that the time instants where \mathfrak{A}_i does while \mathfrak{M} does not make errors when processing x^T are the instants where \mathfrak{M}_i makes errors.

Consequently, if the error of \mathfrak{M} is ρ , and the error of \mathfrak{A}_i is ρ_i , then the error of the circuit comprised of \mathfrak{M} and m assemblies $\mathfrak{M}_1, \dots, \mathfrak{M}_m$ is no greater than $\rho + \sum_{i=1}^m \rho_i$. Thus, selecting memories in assemblies that result in the required error of the entire circuit can be reduced to selecting memories of individual assemblies resulting in appropriate ρ_i values.

Let us now bound the error of circuit \mathfrak{A}_i . This is nothing but the error of circuit \mathfrak{M}_i with the set of input nodes U_0 and the probability for a signal to arrive at $\eta \in U_0$ equal to $p(\eta)$ (noting that the signal can be absent with positive probability). As with the decoder, there is only one type of error—no signal at time t on the output of \mathfrak{M}_i (i.e., “false” signals on output are impossible). Such an error at time t can happen only when at that time a signal appears at a certain $\eta \in U_0$, and in addition, for some $j, 0 \leq j \leq s - 1$, and $\xi \in U_j$ such that for $\eta \in U_j(\xi)$, the elements of ξ are in a nonworking state. The latter, in turn, can happen only when at one of the time instants $t - 1, \dots, t - Q(\xi)$, there was a signal on at least one input from $U_j(\xi)$. The conditional (by the condition that, at time t , there was a signal at a given $\eta \in U_j(\xi)$) probability of the latter event is no greater than $Q(\xi) \sum_{\eta' \in U_j(\xi)} p(\eta')$. Denoting by $\xi_j(\eta)$ the only node of level j that is connected to the node $\eta \in U_0$, the probability of error in \mathfrak{M}_i , provided that at time t there is a signal at $\eta \in U_0$, will not exceed $\sum_{j=0}^{s-1} \left\{ \sum_{\eta' \in U_j(\xi_j(\eta))} p(\eta') \right\} Q(\xi_j(\eta))$. The unconditional probability $p_{\text{err}}(t)$ of error for \mathfrak{M}_i at time t is therefore no greater than $\sum_{\eta \in U_0} p(\eta) \sum_{j=0}^{s-1} \left\{ \sum_{\eta' \in U_j(\xi_j(\eta))} p(\eta') \right\} Q(\xi_j(\eta))$; after straightforward manipulations, this results in

$$(9) \quad p_{\text{err}}(t) \leq \sum_{j=0}^{s-1} \sum_{\xi \in U_j} Q(\xi) p_j^2(\xi),$$

where $p_j(\xi) = p(U_j(\xi)), \xi \in U_j$.

The same reasoning as when constructing UD shows that selecting $Q(\xi)$ according to

$$(10) \quad Q(\xi) = \lfloor \rho_i / s p_j(\xi) \rfloor, \xi \in U_j$$

ensures that $p_{\text{err}}(t) \leq \rho_i$ for all t .

With the above memory selection, (6) and (7) imply that

$$J(\mathfrak{M}_i) \leq n^2 s / \rho_i \leq \rho_i^{-1} n^3.$$

Thus, we have built a circuit \mathfrak{A} that computes $f_{n,m}$ with error $\rho + \sum_{i=1}^m \rho_i$ at the information cost

$$J(\mathfrak{A}) \leq \rho_0^{-1} n^4 + \sum_{i=1}^m \rho_i^{-1} n^3.$$

Note that in the resulting circuit, not only the expected fraction of errors but also the probability of error at every time instant does not exceed $\rho_0 + \sum_{i=1}^m \rho_i$. It remains necessary to optimize the errors $\rho_0, \rho - 1, \dots, \rho_m$ to ensure the total error ρ . Clearly, we should take $\rho_0 = \sqrt{n} \rho / (\sqrt{n} + m), \rho_i = \rho / (\sqrt{n} + m), i = 1, \dots, m$, resulting in $J(\mathfrak{A}) \leq n^3 (\sqrt{n} + m)^2 / \rho$; and for $m \leq n$, we have $J(\mathfrak{A}) \leq 4 \rho^{-1} n^4 m$. The proof is complete.

Abbreviations

DDPF: Direct descendant partition family; GRN: Gene regulatory networks; TPF: Trinomial partition family; TR: Transcription factors; UD: Universal decoder; UGP: Uniform gap partitions

Acknowledgments

We thank Arcadi Nemirovski for inspiring discussions.

Authors' contributions

Conceived and designed the experiments: AG, LS. Performed the experiments: AG, LS, PB. Analyzed the data: AG, PB. Contributed programming tools: LS, PB. Wrote the paper: AG, PB. Performed mathematical modeling: AG, LS, PB. All authors read and approved the final manuscript.

Funding

The authors have no support or funding to report.

Availability of data and materials

The authors confirm that all data underlying the findings are fully available without restriction. All relevant data are within the paper.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Moscow Technological Institute, Moscow, Russia. ²Qcue Inc., Austin, USA. ³Wheely, Moscow, Russia.

Received: 5 April 2019 Accepted: 25 September 2019

Published online: 23 October 2019

References

- Andrews GE (1971) Number Theory
- Andrews GE (1990) Euler's "Exemplum memorabile inductionis fallacis" and q-trinomial coefficients. *J Am Math Soc* 3(3):653–669. <https://doi.org/10.2307/1990932>
- Bocharov P, Goryashko A (2017) New approach to solving partition games. *Appl Math Sci*. <https://doi.org/10.12988/ams.2017.7248>
- Bollobás B, Riordan OM (2004) Mathematical results on scale-free random graphs. In: *Handbook of Graphs and Networks*. <https://doi.org/10.1002/3527602755.ch1>
- Crandall D, Kleinberg J, Suri S, Cosley D, Huttenlocher D (2008) Feedback effects between similarity and social influence in online communities. <https://doi.org/10.1145/1401890.1401914>
- Donetti L, Neri F, Muñoz MA (2006) Optimal network topologies: Expanders, cages, Ramanujan graphs, entangled networks and all that. <http://arxiv.org/abs/0605565>. doi:<https://doi.org/10.1088/1742-5468/2006/08/P08007>
- Erdős P, Rényi A (1960) On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*
- Evans TS (2010) Clique graphs and overlapping communities. *J Stat Mech: Theory and Experiment*. doi:<https://doi.org/10.1088/1742-5468/2010/12/P12037>
- Fortunato S (2010) Community detection in graphs. <http://arxiv.org/abs/0906.0612>. <https://doi.org/10.1016/j.physrep.2009.11.002>
- Gallager R (1970) *Information Theory and Reliable Communication*. Springer, Udine. <https://doi.org/10.1007/978-3-7091-2945-6>
- Goryashko A, Nemirovski A (1978) Estimates of Information Cost of Computing Boolean Functions in Combination Circuits. *Prob of Info Trans* XIV(1):90–100
- Goryashko A, Samokhine L, Bocharov P (2019) Complex Networks and Their Applications VII. In: Aiello LM, Cherifi C, Cherifi H, Lambiotte R, Lió P, Rocha LM (eds). Springer, Cham. pp 553–564
- Helbing D, Brockmann D, Chadefaux T, Donnay K, Blanke U, Woolley-Meza O, Moussaid M, Johansson A, Krause J, Schutte S, Perc M (2014) Saving Human Lives: What Complexity Science and Information Systems can Contribute. *J Stat Phys*. <https://doi.org/10.1007/s10955-014-1024-9>
- Kauffman SA (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*. doi:[http://dx.doi.org/10.1016/0022-5193\(69\)90015-0](http://dx.doi.org/10.1016/0022-5193(69)90015-0). [https://doi.org/10.1016/0022-5193\(69\)90015-0](https://doi.org/10.1016/0022-5193(69)90015-0). [NIHMS150003](https://pubmed.ncbi.nlm.nih.gov/150003/)
- Knuth DE (2011) *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Addison-Wesley Professional
- Kolmogorov AN (1965) Three approaches to the quantitative definition of information. *Prob Inf Trans* 1(1):1–7
- Krioukov D (2014) Brain theory. *Frontiers in Computational Neuroscience* 8(October):114. <https://doi.org/10.3389/fncom.2014.00114>. <http://arxiv.org/abs/1203.2109>
- Leskovec J, Chakrabarti D, Kleinberg J, Faloutsos C, Ghahramani Z (2010) Kronecker graphs: an approach to modeling networks. *J Mach Learn Res* 11:985–1042

- McCarthy P, Benuskova L, Franz EA (2014) The age-related posterior-anterior shift as revealed by voxelwise analysis of functional brain networks. *Front Aging Neurosci*. <https://doi.org/10.3389/fnagi.2014.00301>
- Miller JC, Hagberg A (2011) Efficient Generation of Networks with Given Expected Degrees. In: *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp 115–126. http://link.springer.com/10.1007/978-3-642-21286-4_10
- OEIS Foundation Inc (2018) The On-Line Encyclopedia of Integer Sequences. <https://oeis.org>
- Samokhine L (2017) Trinomial Family Research Toolbox. <https://github.com/samokhine/gory>
- Sorrells TR, Johnson AD (2015) Making sense of transcription networks. <https://doi.org/10.1016/j.cell.2015.04.014>
- Strang A, Haynes O, Cahill ND, Narayan DA (2018) Generalized relationships between characteristic path length, efficiency, clustering coefficients, and density. *Soc Netw Anal Mining*. <https://doi.org/10.1007/s13278-018-0492-3>. 1702.02621
- Whitacre JM (2012) Biological robustness: Paradigms, mechanisms, systems principles. <https://doi.org/10.3389/fgene.2012.00067>
- Yajima S, Inagaki K (1974) Power Minimization Problems of Logic Networks. *IEEE Trans Comput*. <https://doi.org/10.1109/T-C.1974.223878>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
